In Re: U.S.S.N. 10/035,321
Group Art Unit 2144
Rule 132 Declaration of Sivan, Orell and Krupnik, cont'd

17. Similar correspondences may be pointed out between the apparatus and software product claims in the Application and the subject matter of Appendix A.
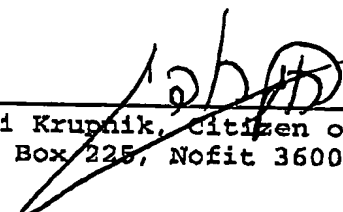
We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

_____   _____
Zohar Sivan, Citizen of Israel   Date
13 Odem Street, Zichron Yaakov, Israel

_____   _____
Dror Orell, Citizen of Israel   Date
42a Givat Downs Street, Haifa 34349, Israel

_____   25 Jan 2006
Hagai Krupnik, Citizen of Israel   Date
P.O. Box 225, Nofit 36001, Israel

8

HotMedia Streaming Servlet APIs proposal

5   This document contains a draft proposal for APIs for a servlet that will support HotMedia content streaming.

The HotMedia Streaming Servlet will be a standalone module running behind/on top of a standard http server. Hence the only interaction with other HotMedia modules is via URL requests that will be sent from HotMedia applet to the http Server. Therefore, the servlet's functionality is to be defined through the format/syntax of

10   URLs which it should respond to, and the content of the http response.

The task of this document is to define this URL syntax. This document does not discuss the changes which are required at the applet side (HotMedia Player) in order to make use of the servlets module. While this does not seem to have impact on the server side it is advised to design the required changes at early stages. (see remark

15   below)

The servlet definition in this document applies only to the first phase of intducing servlets to the HotMedia architecture. This phase is restricted to a "Stateless" implementation. The servlet will not use a "session manager", hence no "Back channel" is provided to support real time/online modification of a previous MVR

20   stream request. However some online capabilities can be achieved by means of closing the current MVR file and reoppening it with new streaming parameters. The question how to use it within the HotMedia Applet should be addressed elsewhere.

If and when a session manager will be added to the servlet module, it will be "backwards compliant". I.e. Previous HotMedia Applet will be able to use the "new"

25   hotMedia Streaming Servlet.

Document version :   0.03 July 15 **Servlet Name :**   We shell refer to this module as *HotMedia Streaming Servlet.*

Servlet Module Requirements:

The servlet module will be a set of pure java classes. The servlet's "main" class name

30   will be hmServlet.class. The "instalation" of this servlet should be simply to put a set of classes in a predefine location within the servlet-engine directory tree.

The servlet module will have to be tested such that it is working properly across main http servers on several platforms. The test plan will be defined later.

Since the servlet has to parse mvr file format it is desirable if the servlet can have a

35   parser which is aligned with the HotMedia Player parser (see remark below)

Servlet URL calls :

To retrieve mvr stream via hmServlet, the URL that will be sent from hmMaster will be of the following Syntax :

http://hostname/servlet/hmServlet/mvrFilePath?queryString

5   where :

- The red part is fixed and hard coded
- hostname is the name of the web server

mvrFilePath is the path to the mvr file including its filename +
10   extension and excluding the root
E.g. if an mvr file can be downloaded from an http server
(without servlet support) using the URL :
http://MyWebServer/MyContentPath/MyMvrFile.mvr
than the servlet URL request will be :

15

http://MyWebServer/servlet/hmServlet/MyContentPath/MyMvr
File.mvr?queryString

- queryString is a string which uses standard SQL format as
described below.
20
- Due to servers inconsistencies - all queries are assumed to be
case sensitive.
- calling http://hostname/servlet/hmServlet/ without MVR
filePath will be used for remote debugging.

25   Using such a call directly from the URL window of a browser
will create a text/html file response with various variables that
will be helpful for debugging (hotMediaServlet version, some
data on the server machine).

- Calling http://hostname/servlet/hmServlet/mvrFilePath without
30   a query string will be used for debugging as well.

In this case the mvr file will be sent to the client "as is", hence
the servlet will serve as an "Identity filter".
This feature can be used later on as a preformance checker to
35   verify that the increase of delay & servers CPU consumption
due to servlet module is minimal.
The http response for these request will be of mime type *"application/octet-stream"*.
This type is "basic" and works well when an applet tries to read binaries input stream
from an http server. To our best knowledge, this detail has no effect on any other
40   component in the framework. (I.e. nothing has to be configured to support this).

2

Servlet queryString :

The servlet queryString is of the following format :

servletAction=action*1*&servletAction=action&param*1*=value*1*&param*2*=value*2*

5   where :

- All sub strings action*i*, param*i* & value*i* should be URL encoded prior to concatenation (see sample code)
- servletAction(s) are the name of the actions that the servlet is requested to perform. Typically only one action will be used

10   per call. However this is not mandatory (see details below).

=> If no action is selected the query string will be null and the mvr file will be sent to the client "as-is".
=> some actions can not be performed simultaneously while

15   some do not (see details below)

- param*i* is the name of the i'th* parameter that is required in order to perform the action (see details below)

* The parameters can be ordered arbitrarily.

20   - value*i* is the value of the i'th* parameter that is required in order to perform the action (see details below)

* The parameters can be ordered arbitrarily.

- There is a delimiter "&" between each parameter/value pair

25   - No spaces will be present in any of the actions/parameters/values unless they are strings which are taken from the mvr headres. In this case - theservlet will support whatever hotMedia applet support.

Suggested Servlet list of action :

30   1. startAtByte
           2. selectTime
           3. selectTracks
           4. skipTracks
           5. selectMediaType
35   6. skipMediaType
           7. FitToBandwidth
           8. limitStreamSpeed
           9. getZoomedImage

3

10. and up - TBD

Note : <u>Not all</u> of the above will be implemented in the coming HotMediaTurbo Beta release.

5

For each action we now specify the parameters that can be used and a list of actions that can not be handled in a single URL request (prohibited simultaneous Actions).

1. **startAtByte** - this action is a "dumb" servlet action which simply start streaming from an arbitrarily requested byte within the MVR file. It is the duty
10   of the hmApplet (hmMaster?) to calculate the start point for the servlet stream and to handle the coming stream accordingly. The output is **not** a valid mvr stream.

    parameters : startByte=intValue   where intValue is a string that represent
15   a non-negative integer which is the location of the first byte within the MVR file.

    prohibited simultaneous Actions :   selectTracks, skipTracks selectMedia, skipMedia, FitToBandwidth,

    output : A bitstream that makes sense only to the sender since it starts from
20   an arbitrary point within the

2. **selectTime** - this action will start streaming from an arbitrarily requested time within a specific track within the MVR file. The output is a valid mvr stream
25   which holds only portion of the track that is selected via the parameters.

    Note that it might be suitable to call the hmPlayer startOfData() following the URL request of this action.

    parameters :   trackId=trackIdValue   where trackIdValue is a string that
30   represent the trackId to be selected
                    startTime=intValue   where intValue is a string that represent a non-negative integer which represents the the requested time in **msec** from which the servlet should stream this track.
                    Note : if startTime parameter is missing from the Query
35   string the servlet will send the bitstream from the beginning of the selected track
                    endTime=intValue   where intValue is a string that represent a non-negative integer which represents the the requested time in **msec** from which the servlet should stop streaming this track.
40   Note : if endTime parameter is missing from the Query string the servlet will send the bitstream till the end of this track

4

prohibited simultaneous Actions : selectTracks, skipTracks selectMedia, skipMedia,

3. **output :** The output is a valid mvr stream which adjust all its headers to correctly define the selected portion of the selected track.

4. **selectTracks** - This action specifically selects the tracks that should be streamed to the client. all other tracks are skipped.

   parameters : trackId=trackIdValue   where trackIdValue is a string that represent the trackId to be selected

   Note : there could be several <trackId=trackIdValue > pairs in a single servlet call.

   prohibited simultaneous Actions : skipTracks, selectMediaType, skipMediaType, startAtByte

   output : A valid mvr stream which is a subset of the original mvr file.

5. **skipTracks** - This action selects the tracks that should be skipped when the mvr file is streamed to the client. All other tracks are sent by their order.

   parameters : trackId=trackIdValue   where trackIdValue is a string that represent the trackId to be selected

   Note : there could be several <trackId=trackIdValue > pairs in a single servlet call.

   prohibited simultaneous Actions : selectTracks, selectMediaType, skipMediaType, startAtByte

   output : A valid mvr stream which is a subset of the original mvr file.

6. **selectMediaType** - This action selects the tracks that corresponds to a specific media type and stream them to the client. All other tracks are skipped.

parameters : MediaType=MediaTypeString where MediaTypeString is a string that represent a hotMedia valid MediaType
Note : there could be several <MediaType=MediaTypeString> pairs in a single servlet call

prohibited simultaneous Actions : selectTracks, skipTracks, skipMediaType, startAtByte

output : A valid mvr stream which is a subset of the original mvr file.

7. **skipMediaType** - This action selects the tracks that should be skipped when the mvr file is streamed to the client. All other tracks are sent by their order.

parameters : MediaType=MediaTypeString where MediaTypeString is a string that represent a HotMedia valid MediaType
Note : there could be several <MediaType=MediaTypeString> pairs in a single servlet call.

prohibited simultaneous Actions : selectTracks, skipTracks, selectMediaType, startAtByte

output : A valid mvr stream which is a subset of the original mvr file.

8. **FitToBandwidth** - This is a recommendation to the servlet to fit the MVR representation to a specific bandwidth. The discussion of how will the server use it is postponed to the future (and so are the parameters).

parameters : TBD

prohibited simultaneous Actions : startAtByte (may be others as well TBD)

output : A valid mvr stream which is a subset of the original mvr file.

6

9. **limitStreamSpeed** - This action recommends to the server to send the stream at a speed no faster than a specified speed.

Typically an http server streams the data As fast as possible. However when the client sends several requests symultanousely (such as request for classes, request for other html pages etc...) and the client's bandwidth is limited it might be desirable to limit the speed of the mvr streaming. This action is provided in order to leave enough bandwidth to other streams that the client opens such as streams to get classes and plain http streams. The output stream content is not changed by this action
Note : At this stage we shell consider this action as exploratory since more has to be done to verify that the client can really benefit from such approach.

parameters : streamSpeed=intValueSpeed where intValueSpeed is a string representation of an integer value which is the preferred speed in kBit/sec

prohibited simultaneous Actions : none

output : The output stream content is not changed by this action

10. **getZoomedImage** - This action provides an interface to high resolution ZoomView of a given Image.

In this implementation it is assumed that :
=> There are two initial images with the same content in different resolution denoted *inline image* and *High resolution image*
=> The two images are of the same aspect-ratio (width/height) up to some rounding error.
=> The client/applet has the capabilities to display the *inline image* as a whole, and then mark a portion of the image of which the user would like to magnify.The look and feel (GUI) of the portion selection is irrelevant to the request as long as the applet is able to describe the selected portion using the parameters described below.

The servlet calculates which portion of the high Resolution it should crop, and how much to down scale it such that the result will have identical range to the inline image crop and identical size to the whole inline image

parameters :
inlineWidth=intValueString - The width (in pixels) of the *Inline Image* to be magnified
inlineHeight=intValueString - The height (in pixels) of the *Inline Image* to be magnified
width=intValueString - The width (in pixels) of the selected portion of the Inline Image to be magnified

7

height=intValueString - The height (in pixels) of the selected portion of the Inline Image to be magnified

x=intValueString - The x coordinate (in pixels) of the upper left corner of the selected portion of the Inline Image to be magnified

y=intValueString - The y coordinate (in pixels) of the upper left corner of the selected portion of the Inline Image to be magnified

imageFilePath=FileString - A path to either a high resolution JPEG image or an MVR file which includes such high-resolution Image.

trackId=trackIdString - Used only if If the High resolution Image is within an MVR file. The trackId of the *High resolution image*

Note : If we will need to fetch the high-resolution images from a Database the imageFilePath parameter will have to be generalize to support the needed info for fetching images from the DataBase. However the issue of handling images that are stored in databases is beyond the scope of this document.

prohibited simultaneous Actions : selectTracks, skipTracks, selectMediaType, skipMediaType, startAtByte,

output : The output stream content is either an MVR stream which has a single media track holding a JPEG image, or a "stripped" JPEG stream (issue TBD)

The dimension of the output image is inline WidthXinlineHeight pixels

Error handling :

When a servlet parses a request it might find out that it can not execute the request. Reasons for this may vary:

- MVR File can not be read (absent/permissions etc...)
- queryString is not valid/can not be handled.
- MVR file has bugs/unsupported features etc....
- runtime errors (outOfMemory, bad connection etc ...)

In terms of error handling, the errors in the servlet can be separated into two types of errors :

1. Errors that the servlet identifies before it starts streaming data to the client.
2. Errors that occur during the servlet's streaming.

1. In the first case the server can signal the problem to the applet using the http header fields.
   For example the httpResponseCode could be changed from the typical 200 to some redefined code number. This can be detected by the applet using URLConnection.getHeaderField(....). Moreover - in this case the servlet can stream an html/text file which describe the problem .... (useful for debugging)

2. In the second case - the only thing that the servlet can do is to close the output stream. This, in turn will cause an exception in the applet.

Additional remarks :

1. We have to analyze the changes needed to be done in the client side at an early stage and make sure that the described URL calls :
    o calls can serve HotMedia applet properly.

    For example : For each request we have to ask were can we fire such a request from the current/modified HotMedia Applet

    o can be created easily within HotMedia Player ?

    For example : For each request we have to ask ourselves how will hotmedia be capable to create the needed parameters for the required action(s).

2. Can we have an mvr parser class that will be common to the applet and the servlet ? Currently the parser is part of hmMasket and it is not trivial to separate its source from the rest of the java file.

A code example :

Here is a sample (pseudo) code that should be integrated into the applet in order to create a URL request to the HotMedia Streaming Servlet
Note : This pseudo code has not been complied and (naturally) not tried as a standalone. Therefore it is not guarantied to be bug free ...

```
String ServletAction[] = .....
String param[] = .....
String value[] = .....
URL mvrURL = new
URL("http://myHost/myContentPath/myMvrFile.mvr");
//                  .......
// absolute & complete URL definition
String servletPath =
mvrURL.getProtocol()+mvrURL.getHost()+"servlet/hmServlet/"+mvrU
RL.getFile();
String queryString="?";
queryString+= URLEncoder.encode("servletAction") +"="
+URLEncoder.encode(ServletAction[i]);
for(.....) {
    queryString+= "&"+ URLEncoder.encode("servletAction") +"="
+URLEncoder.encode(ServletAction[i]);
}
for(.....) {
```

9

```
      queryString+="&" + URLEncoder.encode(param[i]) +"="
+URLEncoder.encode(value[i]);
}
URL url = new URL(servletPath+queryString);
urlc = url.openConnection();
urlc.setUseCaches(false);    // All servlet calls should not be cached in
the browser cache.
dis = new DataInputStream(urlc.getInputStream());
//                    .........
// from here on one can use the dis as if it is a regular dis
from an http response.
```

5

·10

Comments, suggestions and discussions are more than welcomed :
Hagai Krupnik
Audio/Video Group - Multimedia Technologies
IBM Research Laboratory in Haifa, Israel.
LotusNotes : Hagai Krupnik/Haifa/IBM@IBMIL@IBMHAIFA

Phone   : +972-4-8296436
Fax   : +972-4-8296112
snail Mail : MATAM Haifa 31905 ISRAEL.
Email : hagai@il.ibm.com

15

20

APPENDIX B

Hagai
    Krupnik/Haifa/IBM

                                              To
    19:20 15/07/1999   Jeane Chen/Watson/IBM@IBMUS   ,
                              Keeranoor Kumar/Watson/IBM@IBMUS   ,
                              Robert Plotkin/Watson/IBM@IBMUS   ,
                              Subrina Chang/Watson/IBM@IBMUS   ,
                              Liang-Jie Zhang/Watson/IBM@IBMUS   ,
                              James Lipscomb/Watson/IBM@IBMUS   ,
                              Peter Westerink/Watson/IBM@IBMUS   ,
                              Ed Snible/Watson/IBM@IBMUS, Gabriel
                              Taubin/Watson/IBM@IBMUS, Yun
                              Deng/China/IBM@IBMCN, William
                              Gaddy/Watson/Contr/IBM@IBMUS
                                              cc
                              Gilad Cohen/Haifa/IBM@IBMIL, Ehud
                              Karnin/Haifa/IBM@IBMIL@IBMDE, Zohar
                              Sivan/Haifa/IBM@IBMIL
                                          Subject
                              HotMedia Streaming Servlet APIs
                              proposal

Hi all

It's been two weeks since our summit meeting. Personally I've enjoyed the
meeting and it was nice to meet in person people I've only recognize by
their notes header.

Anyway, now we're back to business and there is a "Architecture complete
deadline" in two weeks from now. So....

I have just posted our proposal for HotMedia Streaming Servlet APIs to the
HotMedia discussion database. It is very important that we'll get a serious
feedback on it. Even though, I can guess; most of you are busy with other
aspects of HotMedia/HotMediaTurbo. I just want to make sure that the
designed servlet is really useable for HotMedia applet, otherwise we should
alter the definition before we get into coding. If there is anyone who
can't access this database, please let me know and I'll mail him/her the
file directly.

looking forwards to hear your comments/suggestions

Hagai Krupnik
Audio/Video Group - Multimedia Technologies
IBM Research Laboratory in Haifa, Israel.

LotusNotes : Hagai Krupnik/Haifa/IBM@IBMIL@IBMHAIFA
Email   :   hagai @ il.ibm.com
Phone   : +972-4-8296436
Fax   : +972-4-8296112
snail Mail : MATAM   Haifa 31905   ISRAEL.

APPENDIX C

To: Hagai Krupnik/Haifa/IBM@IBMIL@IBMDE
cc: Jeane Chen/Watson/IBM@IBMUS
From: Subrina Chang/Watson/IBM@IBMUS
Subject: Re: HotMedia Streaming Servlet APIs proposal (Document link:
    Hagai Krupnik(

Hagai,
Thanks for your effort to come up this document. We will need some time to
review your proposal. We will get back to you as soon as possible.

Subrina

Subrina Chang
IBM - Internet Media Group
Tel: 914-784-5756 TieLine: 863-5756
e-mail: subrina@us.ibm.com

APPENDIX D

-Subrina
Chang/Watson/IBM   @
IBMUS   To
                    Hagai Krupnik/Haifa/IBM@IBMIL@IBMDE
                        21:20 26/07/1999   cc
                    Jeane Chen/Watson/IBM@IBMUS, Gilad
                    Cohen/Haifa/IBM@IBMIL@IBMDE, Zohar
                    Sivan/Haifa/IBM@IBMIL@IBMDE   ,
                    Keeranoor Kumar/Watson/IBM@IBMUS
                                    Subject
                    Re: HotMedia Streaming Servlet APIs
                    proposal(Document link: Zohar Sivan
)   Archive   ((

Hagai,
In addition to those servlet actions proposed in your documentation, I like
to introduce more parameters to "startAtByte". Currently, only one
parameter defined for this servlet action - "startByte", an offset to the
MVR file. It may be very straight forward but may not be practical for any
media player. It's hard for a media player to come up with an accurate
offset including header frame and any previous media frames. The following
parameters proposed will give "startAtByte" more flexibility:
 (1)startTime - Instead of a byte offset, a time offset is given if a
specific media track can be measured by time.
 (2)trackID - If 'trackID" is specified, the offset specified in either
"startByte" or "startTime" is only within this specific media track. We can
impose "startTime" to be used with "trackID" in pairs. Only "startByte" can
be left alone. Without any specific track defined, "startByte" interprets
an offset within an entire MVR stream.

As for a common parser for both hmmaster and servlet, it is really nice to
have but not "practical" for HotMedia. There will be one more class file to
be downloaded for HotMedia player. At least, current implementation of mvr
file parser in hmmaster is written in procedures (several functions). It
won't be too difficult to extract them out. Steve Wood in Juerg's area is
the author of hmmaster and also can answer any possible questions in terms
of mvr file parser in details.

Subrina

Subrina Chang
IBM - Internet Media Group
Tel: 914-784-5756 TieLine: 863-5756
e-mail: subrina@us.ibm.com

APPENDIX E

Keeranoor Kumar@IBMUS
09:49 07/28/99PM

To: Arun Ramchandra@Ibmus, Ross Leung/Watson/Ibm@Ibmus, Robert
       Plotkin/Watson/Ibm@Ibmus, Subrina Chang/Watson/Ibm@Ibmus, Liang-Jie
       Zhang/Watson/Ibm@Ibmus, James Lipscomb/Watson/Ibm@Ibmus, Michelle Y
       Kim/Watson/Ibm@Ibmus, Peter Westerink/Watson/Ibm@Ibmus, Steve
       Wood/Watson/Ibm@Ibmus, Jeff Boston/Watson/Ibm@Ibmus, Ed
       Snible/Watson/Ibm@Ibmus, Ehud Karnin/Haifa/Ibm@Ibmil@Ibmde, Leo Y
       Liu/China/Ibm@Ibmcn, Dong Xie/China/Ibm@Ibmcn, Gabriel
       Taubin/Watson/Ibm@Ibmus, Andre Gueziec/Watson/Ibm@Ibmus, Hagai
       Krupnik/Haifa/Ibm@Ibmil, Yun Deng/China/Ibm@Ibmcn, Peter
       Schirling/Burlington/Ibm@Ibmus, Zohar Sivan/Haifa/Ibm@Ibmil, Joshua
       Mittleman/Watson/IBM@IBMUS
cc: Jeane Chen/Watson/IBM@IBMUS, Juerg von Kaenel/Watson/IBM@IBMUS, Jai
       Menon/Watson/Ibm@Ibmus, William Pence/Watson/Ibm@Ibmus
From: Keeranoor Kumar/Watson/IBM @ IBMUS
Subject: HotMedia Turbo Development Plan (Document link: Hagai Krupnik(

Folks,

As some of you know already, I have been asked to oversee the execution of
the HotMedia Turbo development plan. It is a previlege to be working with
such a wonderful and talented team and I am excited. Looking at the chart
that was created at the end of the recent summit, I see 7/30 as the day to
finalize the architecture and 9/30 to build the first function prototype.
Clearly, we need to come up with a much more detailed plan. As a first
step, I am proposing a list of tasks below and some names against each of
them. I welcome your input on the assignment of names and also on whether
you see more task items. Please note that participation in each task is
absolutely open. Please send this ASAP as we should close this list by end
of 7/29 say. The step after that would be to come up with a much more
detailed breakdown for each item and attach a date with them.

    Architecture - steve wood, kumar
        Extension of Extensions API for the new media types - steve wood,
        liang-jie, juerg, kumar
        File format extensions for new media types - peter westerink, jim
        lipscomb, subrina, kumar
            Video - hagai, zohar, subrina
3   D - josh, gabriel, jim lipscomb
            MPEG4 - peter westerink, steve wood
        Stateless Servlet for Content selection - hagai, zohar, subrina,
        liag-jie, kumar
            File format extensions for servlet actions - hagai, zohar, kumar
            Servlet API for client interaction - hagai, zohar, subrina,
            liang-jie
            Servlet design - hagai, zohar
        Authoring extensions for new media types-

```
            Video - hagai, zohar, yun deng
3         D - josh, gabriel
        Authoring tool for MPEG4 - steve wood, peter westerink
    Implementation
        Media Objects
            Video - hagai, zohar, yun deng
3         D  - josh
            MPEG4 - steve wood, peter westerink
        Authoring Extensions
            Video - hagai, zohar, yun deng
3         D - josh
        MPEG4 Authoring Tool (producing an mvr file) - steve wood
        Main Applet & Master Extensions (if any) - steve wood
        Action Handler Extensions - ed snible
        Servlet - hagai, zohar
    Unit Test -  - ed snible (authority for acceptance for integration(
        Media Objects
            Video - hagai, zohar, yun deng
3         D - josh
            MPEG4 - steve wood, peter westerink
        Authoring Extensions
            Video - hagai, zohar, yun deng
3         D - josh
        MPEG4 Authoring Tool
        Mail Applet, Master Extensions & Hyperlinking - steve wood, ed snible
        Servlet - hagai, zohar
    Documentation
    Integration
        Authoring Components - ed snible
        Client Components - ed snible
        Installation Package - ed snible
    System Test
        Test Plan Generation - ed snible, kumar, vee savath
        Test - vee savath's team
```

Please send your response if any ASAP.
Thanks.
-Kumar